



EUROPEAN
SPALLATION
SOURCE

Document Type	Engineering Manual
Document number	ESS-0870816
Date	February 11, 2020
Revision	1
State	Released
Confidentiality Level	Internal
Page	1 (9)

ICS Engineering Manual

FOR FIRMWARE UPDATE OF PCI-BASED MRF PRODUCTS

	Name	Role/Title
Owner	Javier Cereijo Garcia	Timing System Engineer
Author	Javier Cereijo Garcia Jeong Han Lee	Timing System Engineer Senior Controls Engineer

Contents

Contents	2
1 Overview	3
1.1 Scope	3
1.2 Target audience	3
2 System description	3
3 System environment	4
3.1 Hardware	4
3.2 Software	4
4 Engineering procedure	5
4.1 FW update using <code>mrfioc2</code>	5
4.2 FW update using the vendor's own driver	7
Glossary	9
Bibliography	9
Document revision history	9

1 Overview

At European Spallation Source (ESS), Integrated Control System (ICS) uses the MicroResearch Finland (MRF) Timing System¹ as its timing system of the ESS site. The consistent and up-to-date engineering manual is essential for the ESS timing system.

1.1 Scope

- This document explains how to perform a Firmware (FW) update of PCI-based MRF products used at ESS.
- This document identifies two possible ways of performing such update: using the `mrfioc2` EPICS module, or using the vendor's own driver.
- This document attempts to maintain consistency with existing ESS timing system hardware as far as possible.

1.2 Target audience

This document is targeted to ICS engineers and technical stakeholders of the ESS timing system. It is assumed that the target audience has a technical background in the MRF Timing System, the Experimental Physics and Industrial Control System (EPICS) development, and a Linux environment.

2 System description

MRF Technical Reference [see 1, p5] describes the timing system as:

The MRF Timing System provides a complete timing distribution system including timing signal generation with only a few components.

The system is capable of generating and synchronous frequencies, trigger signals and sequences of events, etc. synchronous to an externally provided master clock reference and mains voltage phase signal. Support for timestamps makes the system a global timebase and allows attaching timestamps to collected data and performed actions.

ICS uses and will use the following different types of MRF products:

- MTCA-EVR-300
- PCIe-EVR-300DC

¹<http://www.mrf.fi/>

- MTCA-EVM-300

The scope of this document is to explain how to update the FW of all of them, but other products may also be updated using the same procedure (not tested).

3 System environment

Before describing the engineering procedure for the FW update, it is mandatory to have a proper system environment that consists of specific hardware and software. The information shown in this chapter is used in the ICS lab at ESS.

3.1 Hardware

Table 1 shows the hardware list and its environment. It is possible to use different form factors than what is shown in the examples; for more information check the specific engineering manual.

Hardware	Info
MRF mTCA-EVM-300	
MRF PCIe-EVR-300DC	
MRF mTCA-EVR-300	
μ TCA crate	Incl. PM, MCH
Concurrent Technologies AMC CPU	
Adlink Industrial PC	
Optical cables	LC, optical 850 nm

Table 1 Hardware List and its Environment.

3.2 Software

Table 2 shows the software list and its environment.

Item	Version Info.
CentOS Linux	7.7.1908
Kernel	3.10.0-1062.9.1.el7.x86_64
mrk kernel module	Version: 1 / srcversion E3290AD048B5B57D2EAA55E
EPICS base	7.0.3.1
e3-req	3.1.2
mrfioc2	E3 module ver. 2.2.0-rc7
devLib2	E3 module ver. 2.9.0

Table 2 Software and its version information.

4 Engineering procedure

This chapter provides the instructions on how to perform the FW update.

4.1 FW update using mrfioc2

This is the preferred method, which we have tested with the MTCA-EVR-300(U), PCIE-EVR-300DC and MTCA-EVM-300. Since this method uses the `mrfioc2` module, it is assumed that the ESS EPICS environment (E3) is installed and loaded in the current session². It is also necessary to have the `mrk` kernel module loaded and to register the EVR with the Input/Out Controller (IOC). To do this you need to know the PCI parameters, which can be retrieved as is explained in the engineering manual corresponding to your specific form factor [2, 3, 4].

Then run the following IOC start-up script from the directory containing your bitfile using your PCI and `require` parameters with `iocsh.bash fwupdate.cmd`, which is in `e3-mrfioc/cmds`:

```

1 require mrfioc2,2.2.0-rc7
2 epicsEnvSet("DEV1", "EVR1")
3 mrmEvrSetupPCI("${DEV1}", "0a:00.0")
4 iocInit()
```

Listing 4.1 Start-up script `fwupdate.cmd`. Line 1 and Line 3 should be modified according to the installed `require` version and its corresponding MRF hardware. One

²To install and load E3 check <https://github.com/icshwi/e3> and <https://github.com/icshwi/e3training>.

can get them easily via `bash /tools/get_pciaddr.bash pcieevr300` and make dep within the `e3-mrfioc2` directory.

Before upgrading, it is suggested to backup the existing FW. All Xilinx bit files for a particular device typically have the same size, so you can use the size of the new FW to estimate the size of the old one.

In this example of a PCIe-EVR-300DC with the 207.0 FW, the exact size is 3011417 bytes, which is arbitrarily rounded up to 3MB. Create the backup bit file running from inside the IOC:

```
> flashread("EVR1:FLASH", 0, 0x300000, "PCIe-EVR-300DC.207.0.backup.bit")
| 3080192
| 3014656
| 2949120
| ...
| 65536
| 0

Done
epics>
```

Now write the new FW file, named `PCIe-EVR-300DC-17080207.bit` in this example, that should be located in the current directory. One can check this via `> system("ls *.bit")`:

```
> flashwrite("EVR1:FLASH", 0, "PCIe-EVR-300DC-17080207.bit")
| 0/3011417
| 65536/3011417
| 131072/3011417
| 196608/3011417
| 262144/3011417
| ...
| ...
| 2949120/3011417
| 3011417/3011417

Done
>
```

If the update process is interrupted, **DO NOT POWER CYCLE!** and re-run the update process in order to be completed. After the write completes successfully, power cycle the MRF EVR card to load the new bit file. For more information you can check [\[5\]](#), section Firmware Update, PCIe-EVR-300DC, mTCA-EVR-300].

4.1.1 Troubleshooting

If the `mrfioc2` module was built for a different kernel version than the one the system has, it is possible that the FW update fails while printing a successful message. One can check the kernel module version via `modinfo mrf`. In this case, if the update is interrupted and the board is power cycled, or if you have any other problem that causes an incorrect bit file being flashed, bring the board to ICS for manual flashing.

4.2 FW update using the vendor's own driver

It's better to update the FW with the previous method. This other option is only possible if we can write and compile on the machine, for example with a physical development machine, and we can log in as the root user. For this method you will need to download the driver from <https://github.com/jpietari/mrf-linux-driver>.

First make sure that you power cycle the crate, and that no `mrf` kernel module is loaded. You can check this with `lsmod |grep mrf`. If the module is loaded unload it with `sudo rmmod mrf`. Then clone the driver to the machine with the board you want to update. In this example we work in the `/home/iocuser/FWupdate` directory. The bitfile we are flashing is `mTCA-EVR-300DC-18050207.bit` found in the same directory.

Build the driver:

```
iocuser@cslab-ccpu-crate07: mrf-linux-driver (master)$ make
[...]
```

Log in as root user and install and load the modules:

```
iocuser@cslab-ccpu-crate07: mrf-linux-driver (master)$ sudo su
[sudo] password for iocuser:
[root@cslab-ccpu-crate07 mrf-linux-driver]# make modules_install
make -C /lib/modules/3.10.0-1062.9.1.el7.x86_64/build M=/home/iocuser/FWupdate/mrf-linux-driver
modules_install
make[1]: Entering directory '/usr/src/kernels/3.10.0-1062.9.1.el7.x86_64'
INSTALL /home/iocuser/FWupdate/mrf-linux-driver/pci_mrfevg.ko
Can't read private key
INSTALL /home/iocuser/FWupdate/mrf-linux-driver/pci_mrfevr.ko
Can't read private key
DEPMOD 3.10.0-1062.9.1.el7.x86_64
make[1]: Leaving directory '/usr/src/kernels/3.10.0-1062.9.1.el7.x86_64'
[root@cslab-ccpu-crate07 mrf-linux-driver]# depmod -a
[root@cslab-ccpu-crate07 mrf-linux-driver]# sh module_load
Found 0 Event Generators.
Found 1 Event Receivers.
```

Flash the bitfile to the EVR:

```
[root@cslab-ccpu-crate07 mrf-linux-driver]# dd if=mTCA-EVR-300DC-18050207.bit of=/dev/era1
5881+1 records in
5881+1 records out
3011417 bytes (3.0 MB) copied, 11.2043 s, 269 kB/s
[root@cslab-ccpu-crate07 mrf-linux-driver]#
```

This will take some time. When it's finished power cycle the board to load the new FW.

For more information you can check [6].

Document Type	Engineering Manual
Document number	ESS-0870816
Revision	1

Date	February 11, 2020
State	Released
Confidentiality Level	Internal

4.2.1 Troubleshooting

If the update is interrupted and the board is power cycled, or you have any other problem that causes an incorrect bit file being flashed, bring the board to ICS for manual flashing.

Glossary

Term	Definition
E3	ESS EPICS environment
EPICS	Experimental Physics and Industrial Control System
ESS	European Spallation Source
EVM	Event Master
EVR	Event Receiver
FW	Firmware
ICS	Integrated Control System
IOC	Input/Output Controller
MRF	MicroResearch Finland

Bibliography

- [1] MRF Technical Reference. *Event System with Delay Compensation Technical Reference Firmware 0207*, 9 February 2017.
- [2] Javier Cereijo Garcia and Jeong Han Lee. *ESS-0064948: ICS Engineering Manual for MRF PCIe-EVR-300*, 2020.
- [3] Javier Cereijo Garcia. *ESS-0508480: ICS Engineering Manual for MRF MTCA-EVR-300*, 2020.
- [4] Javier Cereijo Garcia. *ESS-0331569: ICS Engineering Manual for MRF MTCA-EVM-300*, 2020.
- [5] Michael Davidsaver. *EVR Usage Guide*, August 2017, Rev. 8.
- [6] Jukka Pietarinen. *CompactPCI Event Generator and Event Receiver Linux Kernel 2.6 Drivers*, 13 March 2007.

Document revision history

Revision	Reason for and description of change	Author	Date
1	First release	Javier Cereijo Garcia	February 11, 2020